TRBAC의 워크 플로우 과업을 고려한 과업역할할당(TRA) 모델

박동규, 임황빈, 황유동, 정지은
순천향대학교 정보기술공학부

An Task-Role Assignment Model for Workflow Task of TRBAC

Dong-Gue Park, Hwang-Bin Yim, Yu-Dong Hwang, Ji-Eun Jung
Department of Information and Technology, College of Engineering, Soonchunhyang University
E-mail : dgpark@sch.ac.kr, bin9813@yahoo.co.kr, yuri0104@orgio.net, jjsilver@hanmail.net

요 약

역할기반 접근통제(RBAC, Role-Based Access Control)의 개념은 접근통제의 전통적인 강제적 접근통제(MAC, Mandatory Access Control) 및 임의적 접근통제(DAC, Discretionary Access Control)의 대안으로서 많은 관심을 집중시키고 있다. 그러나 역할기반 접근통제를 워크 플로우 시스템에 그대로 적용할 경우에 의무 분리가 복잡해지고, 의무분리를 통해 상호 배타적인 역할을 가지지 못하게 했을 때 서로 관계없는 워크 플로우를 동시에 처리 못하게 되는 문제점이 발생할 수 있다. 이것을 해결하기 위하여 기업 환경의 특성으로 고려한 과업 역할기반 접근제어 모델(TRBAC: Task Role Based Access Control)이 제안되었다. 본 논문에서는 TRBAC의 워크 플로우 과업 관리를 위하여 과업(Task)을 역할(Role)에 할당하는 과업 - 역할할당(TRA)모델을 제안하고 이를 구현한다.

1. 서론

역할기반 접근통제는 관리자에게 편리한 관리 능력을 제공하여 관리업무의 효율성을 꾀할 수 있고, 역할, 역할계층(Role hierarchy), 관계(relationship), 제약(constraint)의 정립을 통하여 사용자의 행동을 정적 또는 동적으로 규제할 수 있으므로 시스템 관리자에게 객체단위가 아닌 추상적인 개념으로 접근을 통제할 수 있어서 실제 환경에 자연스럽게 접목, 구현될 수 있는 장점과 분산환경에서 사용되는 경우 역할기반 접근통제 관리자의 책임을 중앙과 국지 보호 영역으로 구분할 수 있어서 관리책임을 분명히 할 수 있다는 장점이 있다. 그러나 역할기반 접근통제를 워크 플로우 시스템에 그대로 적용할 경우에 의무 분리가 복잡해지고, 의무분리를 통해 상호 배타적인 역할을 가지지 못하게 했을 때 서로 관계없는 워크 플로우를 동시에 처리 못하게 되는 문제점이 발생할 수 있다. 이것을 해결하기 위하여 기업 환경의 특성으로 고려한 과업 역할기반 접근제어 모델(TRBAC: Task Role Based Access Control)이 제안되었다. 또한 시스템에 수많은 사용자, 역할, 권한이 존재하는 경우 한 사람의 보안 관리자가 이들을 모두 관리하는 것은 불가능하므로 역할을 관리하는 관리역할(Administrative Role)을 두어 시스템을 효율적으로 관리할 수 있는 ARBAC(Administrative Role-Based Access Control)이 제안되었다. 그러나 이 방식도 과업을 고려하지 않아 TRBAC의 관리에는 사용하지 못하는 단점이 있다.

본 논문에서는 TRBAC의 관리를 위하여 워크 플로우 과업을 고려한 과업 - 역할 할당(TRA)모델을 제안한다. 제안된 모델은 ARBAC99 모델을 기반으로 각 역할에 할당될 수 있는 과업들을 역할 고유의 과업과 워크 플로우 과업, 상위역할에 상속될 수 있는 과업으로 분리하여 할당하고, 할당 시 전제조건 및 활

본 연구는 정보통신부의 ITRC사업에 의해 수행된 것 임

성화 조건, 유효시간, 동시 최대 활성화 과업 갯수를 둠으로써 의무분리 정책, 및 워크 플로우 특성을 유지할 수 있도록 한다.

2. TRBAC과 ARBAC99

2.1 TRBAC

TRBAC 모델은 객체에 대한 접근권한(permission)이 과업(task)들에 할당이 되고 또 이러한 과업들은 역할(role)에 할당된다. TRBAC 모델에서는 RBAC에서의 일반적인 역할계층(role hierarchy) 대신 감독 - 역할계층(S-RH : supervision role hierarchy)을 사용한다. 일반 역할계층에서는 하위역할의 모든 인가권한이 상위 역할로 계승되지만 감독-역할계층(S-RH)에서는 하위 역할의 인가권한 중 계승(inheritance)성질이 있는 권한에 해당하는 과업의 인가권한만 상위로 계승된다.

위에서 설명한 TRBAC 모델을 예를 들면 다음그림 1과 같다.

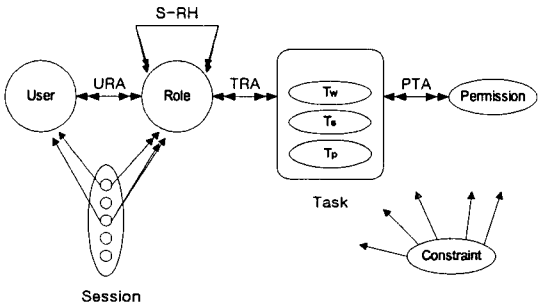

그림 1 T-RBAC 모델의 예

위 그림 1에서 과업을 T로 정의하고 다른 역할들에 할당할 수 있는 과업을T_s, 다른 역할들에 할당할 수 없고 특정역할에만 할당되는 과업을 T_p로 표현하였다. T_w에 속하는 과업은 인스턴스가 활성화 될 때 인가권한도 활성화되고, 과업 인스턴스가 비활성화되면 인가권한도 비활성화 된다. T_w의 과업들은 행위 기반 접근제어를 가능하게 하기 위해 과업의 활성화 조건(activation condition), 활성화 된 뒤의 유효시간(time constraint), 동시에 활성화 할 수 있는 최대 과업 인스턴스의 수(cardinality)를 속성값으로 갖는다.

2.2 ARBAC99

ARBAC는 사용자에게 역할을 할당하는 URA (User - Role Assignment)와 권한에 역할을 할당하는 PRA(Permission-Role Assignment), 역할에 역할을 할당하는 RRA(Role-Role Assignment) 모델로 이루어져 있다. 다음 그림 2는 역할계층과 관리역할계층의 일부를 예로 보여준다.

ARBAC99에서는 역할을 이동자격(mobile membership)과 부동자격(immobile membership)으로 분류한다. 역할에 대해 이동자격이 주어진 사용자는 관리역할이 다른 역할을 사용자에게 매핑 시킬 때 다른 역할로 이동시킬 수 있음을 의미하고, 역할에 대해 부동자격이 주어진 사용자는 관리 역할이 다른 역할을 사용자에게 매핑 시킬 때 다른 역할로 이동시킬 수 없음을 의미한다.

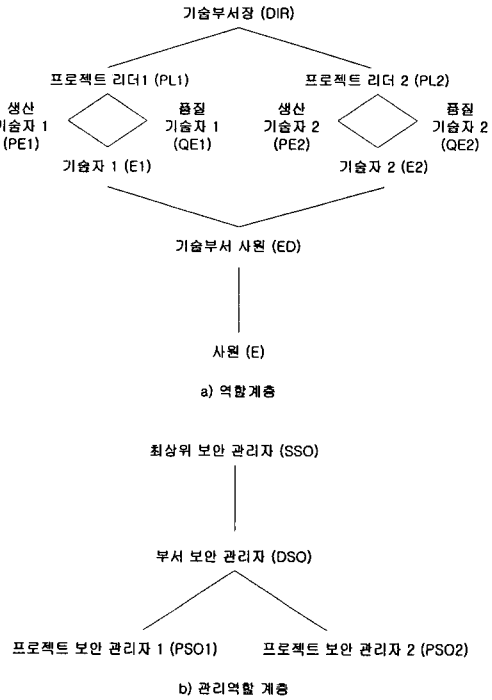

그림 2 역할계층과 관리역할계층의 예

3. TRA

위 그림1의 TRBAC의 모델과 그림 2의 역할과 관리역할을 이용하여 TRA에서 can-assign 관계에 의한 과업-역할 할당의 예를 들면 표 1,2,3,4와 같다.

표 1 can-assignT_s-M의 예

관리 역할	전제 역할	역할 범위
DSO	DIR_s	[$PL1_s$, $PL1_s$]
DSO	DIR_s	[$PL2_s$, $PL2_s$]
PSO1	$PL1_s \wedge \overline{QE1}$	[$PE1_s$, $PE1_s$]
…	…	…

표 2 can-assignT_s-IM의 예

관리 역할	전제 역할	역할 범위
DSO	DIR_s	[$PL1_s$, $PL1_s$]
DSO	DIR_s	[$PL2_s$, $PL2_s$]
PSO1	$PL1_s \wedge \overline{QE1}$	[$PE1_s$, $PE1_s$]
…	…	…

표 3 can-assignT_p의 예

관리역할	전제 역할	역할 범위
DSO	$\overline{DIR} \wedge \overline{PL2_p} \wedge \overline{PE1_p} \wedge \overline{QE1_p} \wedge \overline{PE2_p} \wedge \overline{QE2_p} \wedge \overline{E1_p} \wedge \overline{E2_p} \wedge \overline{ED_p} \wedge \overline{E_p}$	[$PL1_p$, $PL1_p$]
DSO	$\overline{DIR} \wedge \overline{PL1_p} \wedge \overline{PE1_p} \wedge \overline{QE1_p} \wedge \overline{PE2_p} \wedge \overline{QE2_p} \wedge \overline{E1_p} \wedge \overline{E2_p} \wedge \overline{ED_p} \wedge \overline{E_p}$	[$PL2_p$, $PL2_p$]
PSO1	$\overline{DIR} \wedge \overline{PL1_p} \wedge \overline{PL2_p} \wedge \overline{QE1_p} \wedge \overline{PE2_p} \wedge \overline{QE2_p} \wedge \overline{E1_p} \wedge \overline{E2_p} \wedge \overline{ED_p} \wedge \overline{E_p}$	[$PE1_p$, $PE1_p$]
…	…	…

표 4 can-assignT_w의 예

관리 역할	전제 역할	역할 범위	활성화 조건	유효 시간	cardinality
DSO	$\overline{DIR} \wedge \overline{PL2} \wedge \overline{PE1_w} \wedge \overline{QE1_w} \wedge \overline{PE2_w} \wedge \overline{QE2_w} \wedge \overline{E1_w} \wedge \overline{E2_w} \wedge \overline{ED_w} \wedge \overline{E_w}$	[$PL1_w$, $PL1_w$]	AC1	DT1	C1
DSO	$\overline{DIR} \wedge \overline{PL1_w} \wedge \overline{PE1_w} \wedge \overline{QE1_w} \wedge \overline{PE2_w} \wedge \overline{QE2_w} \wedge \overline{E1_w} \wedge \overline{E2_w} \wedge \overline{ED_w} \wedge \overline{E_w}$	[$PL2_w$, $PL2_w$]	AC2	DT2	C2
PSO1	$\overline{DIR} \wedge \overline{PL1_w} \wedge \overline{PL2_w} \wedge \overline{QE1_w} \wedge \overline{PE2_w} \wedge \overline{QE2_w} \wedge \overline{E1_w} \wedge \overline{E2_w} \wedge \overline{ED_w} \wedge \overline{E_w}$	[$PE1_w$, $PE1_w$]	AC3	DT3	C3
…	…	…	…	…	…

위의 표 1과 2는 여러 역할들에 할당될 수 있는 과업 T_s에 대한 과업-역할 할당의 예로 역할을 이동자격(mobile membership)과 부동자격(immobile member- ship)으로 분류하여 표현하였다. 표 1과 2에서 관리역할은 전제역할에 할당된 공유 과업을 역할범위에 해당하는 역할에 할당할 수 있음을 알 수 있다. 관리역할 DSO는 역할 DIR에 할당된 과업 T_s를 역할 PL1에 할당할 수 있다는 의미이다.

위의 표 3은 다른 역할들에 할당할 수 없고 특정역할에만 할당되는 과업 T_p에 대한 과업-역할 할당의 예로 특정역할에만 과업이 할당되므로 이동자격과 부동자격으로 분류가 불가능하다. 과업 T_p는 역할 범위에 해당하는 특정 역할에만 할당된다는 것을 알 수 있다. 과업 T_p는 특정역할에만 과업이 할당되므로 이동자격과 부동자격으로 분류가 불가능하다.

위의 표 4는 과업 T_w에 대한 과업-역할 할당의 예를 보여준다. 과업 T_w는 T_p와 같이 특정역할에만 과업이 할당된다. 표 4의 활성화 조건은 과업 T_w가 활성화되기 위한 조건을 의미한다. 활성화 조건의 예를 들면 신입 사원의 경우 역할 E가 할당되었을 때 "역할이 할당된 날로부터 3일 이내"를 들 수 있다. 유효시간은 역할이 할당된 후 과업-역할 할당 관계가 지속되는 시간을 의미하며 cardinality는 동시에 활성화할 수 있는 과업-역할 할당 관계를 의미한다.

4. 구현

TRA를 구현하기 위하여 다음 그림 3과 같은 ER 다이어그램을 작성하였다.

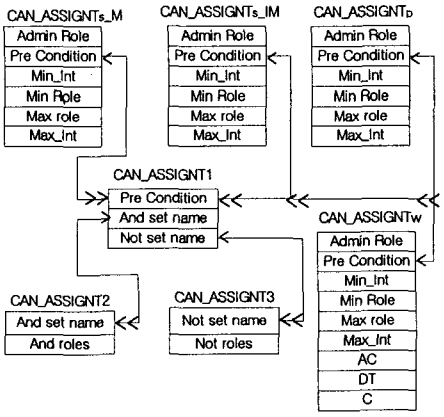

그림 3 can-assignT_s-M과 can-assignT_s-IM, can-assignT_p, can-assignT_w 관계의 ER 다이어그램

그림 3은 can-assignT_s-M과 can-assignT_s-IM, can-assignT_p , can-assignT_w 의 경우이며 can-revokeT_s-M과 can-revokeT_s-IM, can-revokeT_p , can-revokeT_w도 그림 3과 같은 형식의 ER 다이어그램으로 작성할 수 있다.

다음의 표 5, 6, 7, 8, 9는 그림 3의 ER 다이어그램을 이용하여 테이블을 작성하고 작성된 테이블에 관리역할에 대한 데이터를 입력한 예이다. can-assignT2와 can-assignT3 테이블도 표6과 같은 형식으로 작성할 수 있다.

표 5 can_assignT_s_M 테이블의 예

AR	PC	Min_Int	Min_R	Max_R	Max_Int
DSO	C1	[	$PL1_s$	$PL1_s$	]
DSO	C1	[	$PL2_s$	$PL2_s$	]
PSO1	C2	[	$PE1_s$	$PE1_s$	]
…	…	…	…	…	…

표 6 can_assignT_s_IM 테이블의 예

AR	PC	Min_Int	Min_R	Max_R	Max_Int
DSO	C1	[	$PL1_s$	$PL1_s$	]
DSO	C1	[	$PL2_s$	$PL2_s$	]
PSO1	C2	[	$PE1_s$	$PE1_s$	]
…	…	…	…	…	…

표 7 can_assignT_p 테이블의 예

AR	PC	Min_Int	Min_R	Max_R	Max_Int
DSO	C11	[	$PL1_p$	$PL1_p$	]
DSO	C12	[	$PL2_p$	$PL2_p$	]
PSO1	C21	[	$PE1_p$	$PE1_p$	]
…	…	…	…	…	…

표 8 can_assignT_w 테이블의 예

AR	PC	Min _Int	Min _R	Max _R	Max _Int	AC	DT	C
DSO	C11	[	$PL1_p$	$PL1_p$	]	AC1	DT1	C1
DSO	C12	[	$PL2_p$	$PL2_p$	]	AC2	DT2	C2
PSO 1	C21	[	$PE1_p$	$PE1_p$	]	AC3	DT3	C3
…	…	…	…	…	…	…	…	…

표 9 can_assignT1 테이블의 예

PC	and_set_name	not_set_name
C1	ASET1	null
C2	ASET2	NSET1
C3	ASET2	NSET2
…	…	…

본 논문에서는 TRA를 구현하기 위해 Oracle8i 에서 SQL을 확장한 PL/SQL을 사용하여 권한-역할 할당을 위한 관리도구를 저장 프로시저로 작성하였다. 작성한 저장 프로시저는 다음과 같다.

- share_assignT(role, ttask, arole, mobile)
- share_weak_revokeT(role, ttask, arole, mobile)
- share_strong_revokeT(role,ttask, arole, mobile)
- private_assignT(role, ttask, arole)
- private_revokeT(role, ttask, arole)
- workflow_assignT(role, ttask, arole, ac, dt, c)
- workflow_revokeT(role, ttask, arole)

저장 프로시저는 역할에 과업을 할당하고 과업을 취소하는 기능을 한다.

파라미터 role은 과업이 할당되거나 취소될 역할이고 ttask(target task)는 역할에 할당되거나 취소될 과업이다. arole(administrative role)은 role에 ttask 과업을 할당하거나 취소할 관리역할이다. mobile 파라미터는 ttask에 입력되는 과업이 이동자격을 가지는지 부동자격을 가지는지를 표시한다. mobile 파라미터에 입력된 값이 이동자격을 표시하면 저장 프로시저는 can-assignT_s-M 테이블에서 관리역할과 역할의 범위를 검색하고 부동자격을 표시하면 저장 프로시저는 can-assign_IM 테이블에서 관리역할과 역할의 범위를 검색한다. 과업 저장 프로시저 중 share_assignT, share_weak_revokeT, share_strong_revokeT는 여러 역할들에 할당될 수 있는 과업 T_s에 대한 과업-역할 할당과 취소 저장 프로시저이고 private_ assignT, private_revokeT는 다른 역할들에 할당할 수 없고 특정역할에만 할당되는 과업 T_p에 대한 과업-역할 할당과 취소 저장 프로시저이다.

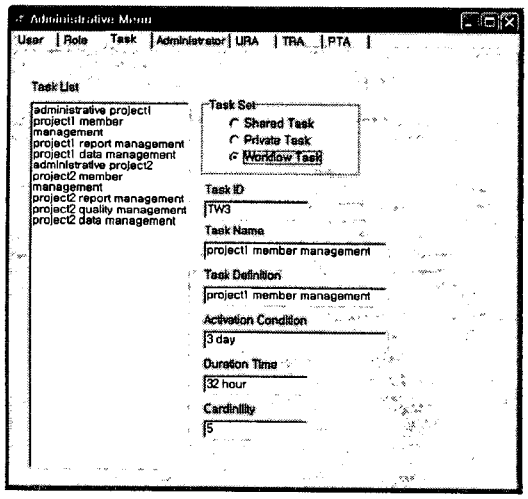

그림 4 과업 정의 화면

workflow_assignT, workflow_revokeT는 과업 T_w에 대한 과업-역할 할당 취소 저장 프로시저이다. ac는 role에 할당된 ttask가 활성화 될 수 있는 조건을 의미하며, dt는 활성화된 과업이 유지될수 있는 시간을 의미하고, c는 동시에 활성화 될 수 있는 과업의 최대 개수를 의미한다.

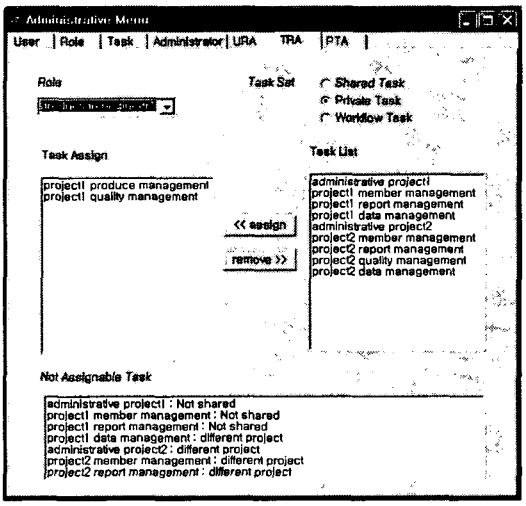

그림 5 과업 - 역할 할당 화면

위 그림 4, 5는 저장 프로시저를 호출해서 사용할 수 있는 관리도구의 GUI이다.

5. 결론

시스템에 수많은 사용자, 역할, 권한이 존재하는 경우 이를 관리하는 관리역할을 두어 시스템을 효율적으로 관리할 수 있는 방법(ARBAC)이 필요하다. 또한 워크 플로우 시스템에서 과업을 고려한 접근제어 방식으로 TRBAC 이 제안 되었다.

본 논문에서는 TRBAC의 워크 플로우 과업 관리를 위하여 과업(Task)을 역할(Role)에 할당하는 과업 - 역할할당(TRA)모델을 제안한다. 제안된 모델은 ARBAC99 모델을 기반으로 각 역할에 할당될 수 있는 과업들을 역할 고유의 과업과 워크 플로우 과업, 상위역할에 상속될 수 있는 과업으로 분리하여 할당하고, 할당 시 전제조건 및 활성화 조건, 유효시간, 동시 최대 활성화 과업 갯수를 둠으로써 의무분리 정책, 및 워크 플로우 특성을 유지할 수 있도록 한다.

본 논문에서는 TRBAC에서 ARBAC99를 기반으로 워크 플로우 과업을 고려한 과업 - 역할 할당 관리도구를 구현하였다. 구현된 관리도구는 볼랜드의 델파이로 개발하였으며 데이터베이스로는 오라클의 저장 프로시저를 사용하였다. 구현된 관리도구를 실제 기업환경의 시스템에 적용해 보고 향후 워크 플로우 시스템의 접근제어를 위한 범용 컴포넌트를 구현하고자 한다.

[참고문헌]

[1] Ravi Sandhu, Venkata Bhamidipati, "Role-Based Administration of User-Role Assignment : The URA97 Model and its Oracle Implementation", Journal of Computer Security, Volume 7,1999

[2] Ravi Sandhu , Qamar Munawer, "The ARBAC99 Model for Administration of Roles", ACSAC, 1999

[3] Ravi Sandhu , Venkata Bhamidipadi, "An Oracle Implementation of the PRA97 Model for Permission-Role Assignment ", ACM RBAC, 1998

[4] Ravi Sandhu, Joon S. Park, "Decentralized User-Role Assignment for Web-based Intranets ", ACM RBAC, 1998

[5] 오세종, 박석. "역할기반 접근제어에서 기업 환경에 적합한 역할계층의 구성에 관한 연구", 한국 통신정보보호학회 종합학술발표회 논문집, Vol9, No1, 1999,11.

[6] 오세종, 박석. "기업환경을 위한 과업-역할기반 접근제어 모델", 한국통신정보보호학회논문지, 제 11권, 제1호, 2001,2.

[7] 황유동, 박동규, "TRBAC의 관리를 위한 과업역할 할당(TRA) 모델", CMC2001, 2001.6